AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of processing data with one or more data processing threads executing under control of a first operating system and one or more data processing threads executing under control of a second operating system, said method comprising the steps of:

executing a first data processing thread under control of said first operating system;

receiving a first interrupt operable to suspend execution of said first data processing thread at a first thread exit point and to trigger interrupt processing under control of said second operating system;

executing interrupt handing code under control of said second operating system;

generating a return interrupt;

determining under control of said second operating system a return data processing thread to be executed under control of said first operating system;

executing return interrupt handling code under control of said first operating system, said return interrupt handling code being ~~operable~~configured such that:

(i) if said return data processing thread is said first data processing thread, then said first data processing thread is resumed at said first thread exit point; and

(ii) if said return data processing thread is a second data processing thread different from said first data processing thread, then a thread switching operation is performed under control of said first operating system to save context data associated with said first data

processing thread at said first thread exit point such that said first data processing thread may later be resumed from said first thread exit point and to trigger execution of a second data processing thread under control of said first operating system.

2. (Original) A method as claimed in claim 1, wherein said first operating system executes in a secure domain and said second operating system executes in a non-secure domain, wherein a data processing thread executing in said secure domain has access to secure data which is not accessible to a data processing thread executing in said non-secure domain.

3. (Original) A method as claimed in claim 1, wherein said return interrupt is a software simulated interrupt.

4. (Original) A method as claimed in claim 3, wherein said software simulated interrupt includes a branch instruction triggering a branch in program execution flow to said return interrupt handling code.

5. (Original) A method as claimed in claim 1, wherein said return interrupt specifies one or more of:

a data processing thread executing under control of said second operating system when said return interrupt was generated; and

one or more parameter values.

6. (Original) A method as claimed in claim 1, wherein switches between processing under control of said first operating system and processing under control of said second operating system take place via a monitor mode of operation executing a monitor mode program.

7. (Original) A method as claimed in claim 1, wherein triggering execution of said second data processing thread is one of:

starting execution of said second data processing thread as new data processing thread; and

resuming execution of said second data processing thread following an earlier suspension of said second data processing thread.

8. (Currently Amended) Apparatus for processing data with one or more data processing threads executing under control of a first operating system and one or more data processing threads executing under control of a second operating system, said apparatus comprising:

execution logic ~~operable~~configured to execute a first data processing thread under control of said first operating system;

interrupt receiving logic ~~operable~~configured to receive a first interrupt and suspend execution of said first data processing thread at a first thread exit point and to trigger interrupt processing under control of said second operating system;

interrupt handling execution logic ~~operable~~configured to execute interrupt handing code under control of said second operating system;

a return interrupt generator ~~operable~~configured to generate a return interrupt;

return data processing thread determining logic ~~operable~~configured to determine under control of said second operating system a return data processing thread to be executed under control of said first operating system;

return interrupt handling code execution logic ~~operable~~configured to execute return interrupt handling code under control of said first operating system, said return interrupt handling code being ~~operable~~configured such that:

(i) if said return data processing thread is said first data processing thread, then said first data processing thread is resumed at said first thread exit point; and

(ii) if said return data processing thread is a second data processing thread different from said first data processing thread, then a thread switching operation is performed under control of said first operating system to save context data associated with said first data processing thread at said first thread exit point such that said first data processing thread may later be resumed from said first thread exit point and to trigger execution of a second data processing thread under control of said first operating system.

9. (Original) Apparatus as claimed in claim 8, wherein said first operating system executes in a secure domain and said second operating system executes in a non-secure domain, wherein a data processing thread executing in said secure domain has access to secure data which is not accessible to a data processing thread executing in said non-secure domain.

10. (Original) Apparatus as claimed in claim 8, wherein said return interrupt is a software simulated interrupt.

11. (Original) Apparatus as claimed in claim 10, wherein said software simulated interrupt includes a branch instruction triggering a branch in program execution flow to said return interrupt handling code.

12. (Original) Apparatus as claimed in claim 8, wherein said return interrupt specifies one or more of:

a data processing thread executing under control of said second operating system when said return interrupt was generated; and

one or more parameter values.

13. (Original) Apparatus as claimed in claim 8, wherein switches between processing under control of said first operating system and processing under control of said second operating system take place via a monitor mode of operation executing a monitor mode program.

14. (Original) Apparatus as claimed in claim 8, wherein triggering execution of said second data processing thread is one of:

starting execution of said second data processing thread as new data processing thread; and

resuming execution of said second data processing thread following an earlier suspension of said second data processing thread.

15. (Currently Amended) A computer program product having a computer program ~~operable~~configured to control a data processing apparatus in accordance with a method as claimed in claim 1.